

Министерство образования и науки Украины
Севастопольский национальный технический университет



ОБРАБОТКА МАССИВОВ

Методические указания
к выполнению лабораторной работы № 6
по дисциплине
«Алгоритмические языки и программирование»
для студентов направления подготовки
0914 «Компьютеризированные системы, автоматика и управление»
дневной формы обучения

Севастополь
2007



СОДЕРЖАНИЕ

1. Лабораторная работа №6 Обработка массивов	3
1.1. Цель работы	3
1.2. Краткие теоретические сведения	3
1.3. Пример программы	3
1.4. Задание на работу	4
1.5. Содержание отчета и порядок защиты работы	6
1.6. Контрольные вопросы	7

1. ЛАБОРАТОРНАЯ РАБОТА №6

ОБРАБОТКА МАССИВОВ

1.1. Цель работы

Составление циклических алгоритмов и программ, осуществляющих типичные операции над одномерными массивами. Приобретение навыков программирования ввода и вывода массивов.

1.2. Краткие теоретические сведения

В программе на языке Си можно использовать структурированные типы данных. К ним будем относить массивы, структуры и файлы.

Массив состоит из многих элементов одного и того же типа. Ко всему массиву целиком можно обращаться по имени. Кроме того, можно выбирать любой элемент массива. Для этого необходимо задать индекс, который указывает его относительную позицию. Число элементов массива назначается при его определении и в дальнейшем не меняется. Если массив объявлен, то к любому его элементу можно обратиться следующим образом: указать имя массива и индекс элемента в квадратных скобках. Массивы определяются так же, как и переменные:

```
int a[100];
char b[20];
float d[50];
```

В первой строке объявлен массив *a* из 100 элементов целого типа: *a*[0], *a*[1], ..., *a*[99] (индексация всегда начинается с нуля). Во второй строке элементы массива *b* имеют тип *char*, а в третьей - *float*.

1.3. Пример программы

Следующая программа создает массив из 100 случайных целых чисел. Диапазон значений элементов массива от -50 до 50.

Функция *ChangeNegative()* заменяет все отрицательные элементы массива нулевыми значениями. Входными данными для функции должны быть массив и количество его элементов. Поскольку имя массива представляет собой указатель на его нулевой элемент, в функцию массив передается через указатель. Количество элементов в массиве должно передаваться отдельным параметром, потому что, в отличие от строк символов, использующих признак конца строки, для массивов общего вида никакого признака конца массива не существует.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
void ChangeNegative(int *arr, int arrSize);
int main (void)
{
    const int n=50;
```

```

int a[n];
int i;
randomize();
for (i=0; i<n; a[i++]=random(101)-50);
printf("Array:\n");
for (i=0; i<n; i++)
{
    printf("%d\t",a[i++]);
    if(!((i+1)%10)) printf("\n");
}
ChangeNegative(a,n);
printf("Changed Array:\n");
for (i=0; i<n; i++)
{
    printf("%d\t",a[i++]);
    if(!((i+1)%10)) printf("\n");
}
return 0;
}
void ChangeNegative(int *arr, int arrSize)
{
    int i;
    for (i=0; i<arrSize; i++)
        if (arr[i]<0) arr[i]=0;
}

```

1.4. Задание на работу

- 1) Изучите пример программы, приведенный в разделе 1.2.
- 2) Получите у преподавателя номер варианта индивидуального задания на лабораторную работу. Варианты заданий приведены в таблице 1.1.
- 3) Создайте программу, реализующую удовлетворять следующим требованиям:

– В основной функции должен объявляться массив целых чисел и заполняться случайными значениями. Количество элементов массива и диапазоны значений указаны в таблице 1.1.

– Основная функция должна вызывать две вспомогательные функции. Первая функция должна быть предназначена вычисления значения в соответствии с заданием указанным в столбце 4 таблицы вариантов. Вторая функция должна быть предназначена для сортировки массива в соответствии с заданием указанным в столбце 5 таблицы вариантов;

– Программа должна осуществлять вывод на экран исходного массива, результата работы первой функции с комментариями, отсортированного массива.

Таблица 1.1 – Варианты заданий

№ варианта	Задания и значения исходных данных			
	Количество элементов	диапазон значений	Первая функция должна вычислять	Вторая функция. Метод и направление сортировки
1	2	3	4	5
1	10	-100..100	Сумму положительных элементов массива	Прямое включение, по возрастанию
2	20	0..100	Минимальный элемент массива	Прямой выбор, по возрастанию
3	30	-100..100	Максимальный элемент массива	Прямой обмен, по возрастанию
4	40	-50..50	Количество элементов массива, больших A	Прямое включение, по убыванию
5	50	0..200	Сумму первых m элементов	Прямой выбор, по убыванию
6	60	0..50	Произведение элементов с номерами от k до m	Прямой обмен, по убыванию
7	70	-100..100	Сумму модулей элементов массива	Прямое включение, по возрастанию
8	80	0..100	Количество элементов массива, меньших A	Прямой выбор, по возрастанию
9	90	-100..100	Сумму отрицательных элементов массива	Прямой обмен, по возрастанию
10	100	-50..50	Минимальный положительный элемент массива	Прямое включение, по убыванию
11	10	0..200	Сумму последних m элементов	Прямой выбор, по убыванию
12	20	0..50	Сумму элементов массива, больших A	Прямой обмен, по убыванию
13	30	-100..100	Максимальный отрицательный элемент массива	Прямое включение, по возрастанию
14	40	0..100	Сумму элементов массива с четными номерами	Прямой выбор, по возрастанию
15	50	-100..100	Количество элементов массива, по модулю больших A	Прямой обмен, по возрастанию
16	60	0..100	Номер максимального элемента массива	Прямое включение, по убыванию
17	70	-100..100	Номер минимального положительного элемента массива	Прямой выбор, по убыванию

Таблица 1.1 – Варианты заданий

№ варианта	Задания и значения исходных данных			
	Количество элементов	диапазон значений	Первая функция должна вычислять	Вторая функция. Метод и направление сортировки
1	2	3	4	5
18	80	-50..50	Первый отрицательный элемент массива	Прямой обмен, по убыванию
19	90	0..200	Сумму элементов массива с нечетными номерами	Прямое включение, по возрастанию
20	100	0..50	Номер минимального элемента массива	Прямой выбор, по возрастанию
21	10	-100..100	Первый положительный элемент массива	Прямой обмен, по возрастанию
22	20	0..100	Количество элементов массива, равных A	Прямое включение, по убыванию
23	30	-100..100	Номер максимального по модулю элемента массива	Прямой выбор, по убыванию
24	40	-50..50	Минимальный по модулю элемент массива	Прямой обмен, по убыванию
25	50	0..200	Количество элементов массива, равных 0	Прямое включение, по возрастанию

Примечание: Значения A, m, k должны быть параметрами функции и должны вводиться с клавиатуры в основной функции

1.5. Содержание отчета и порядок защиты работы

Выполнение и защита лабораторной работы производится каждым студентом индивидуально. Защита результатов лабораторной работы осуществляется при наличии работающей программы и полностью оформленного отчета.

Отчет должен включать в себя следующие разделы

- титульный лист;
- цель работы;
- постановка задачи. Этот раздел должен содержать вариант задания в соответствии с таблицей 1.1;
- схема программы. Этот раздел должен содержать три схемы: схему основной функции main, схемы обоих вспомогательных функций;
- текст программы на языке C;
- результаты работы программы;
- выводы.

Защита работы состоит в следующем:

- представление работающей программы на компьютере;
- предъявление отчета, оформленного в соответствии с указанными требованиями;

ответы на вопросы преподавателя по теоретической и практической части работы. Примеры возможных вопросов приведены в подразделе 1.5.

1.6. Контрольные вопросы

- 1) Какие типы данных относятся к простым, а какие – к структурированным?
- 2) Что представляет собой массив? Каковы основные характеристики массива?
- 7) Изобразите синтаксис объявления типа-массива.
- 8) Перечислите операции, допустимые для переменных-массивов.
- 9) Как осуществляется доступ к элементам массива?
- 10) Что такое индекс массива? Какие допустимы типы индексов массива?
- 11) Изложите алгоритм поиска максимального элемента в массиве.